Information theory

Kevin P. Murphy

Last updated October 25, 2006

* Denotes advanced sections that may be omitted on a first reading.

1 Introduction

Information theory is concerned with two main tasks. The first task is called **data compression (source coding)**. This is concerned with removing redundancy from data so it can be represented more compactly (either exactly, in a lossless way, or approximately, in a lossy way). The second task is **error correction (channel coding)**, which means encoding data in such a way that it is robust to errors when sent over a noisy channel, such as a telephone line. It turns out that the amount by which you can compress data, and the amount of redundancy you need to add to a message before transmission, are both closely related to how predictable the source of data is, i.e., to its probability distribution. Hence there is a deep connection between information theory and statistics/ machine learning (indeed, there is an excellent book on this topic [Mac03]). In this chapter, we introduce some of the key information-theoretic measures of uncertainty and statistical dependence, such as entropy and mutual information.

2 Basic concepts

2.1 Data encoding

Suppose we want to encode some data, such as a text document. How much space (i.e., number of bits) does this take? It depends on the encoding scheme which we use for words, and on how often each word occurs (on average).

Let us consider a simple example. Suppose we want to encode a "sentence" that consists of the 5 letters $\{a, b, c, d, e\}$. We need $\lceil \log_2 5 \rceil = 3$ bits to represent each letter. A simple encoding scheme is as follows:

$$a \rightarrow 000, b \rightarrow 001, c \rightarrow 010, d \rightarrow 011, e \rightarrow 100 \tag{1}$$

So the string "abcd" gets encoded as

$$abcd \rightarrow 000, 001, 010, 011 \tag{2}$$

It is clear that we need $3N$ bits to encode a string of length N. (Note that to decode this, it is critical that we know each codeword has length 3, since in practice we write the bit string without the commas. We will discuss this more below.)

Now suppose some letters are more common than others. In particular, suppose we have the distribution

$$p(a) = 0.25, p(b) = 0.25, p(c) = 0.2, p(d) = 0.15, p(e) = 0.15 \tag{3}$$

Intuitively, we can use fewer bits by assigning short codewords, such as 00 and 10, to common letters such as a and b, and long codewords, such as 011, to rare letters such as e. In particular, consider the following encoding scheme:

$$a \rightarrow 00, b \rightarrow 10, c \rightarrow 11, d \rightarrow 010, e \rightarrow 011 \tag{4}$$

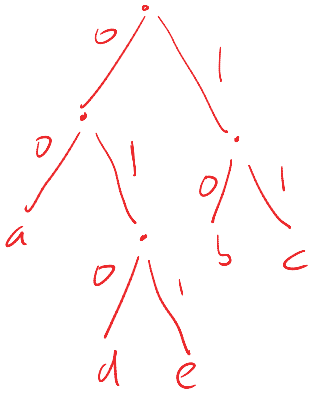


Figure 1: An illustration of a prefix-free code.

This encoding scheme is **prefix-free**, which means that no codeword is a prefix of any other: see the tree in Figure 1. Hence when we concatenate any sequence of codewords, we can uniquely decode the answer by repeatedly traversing paths in the tree, e.g.

$$001011010 \rightarrow 00, 10, 11, 010 \rightarrow abcd \tag{5}$$

The average number of bits that are used by this encoding scheme is given by

$$0.25 * 2 + 0.25 * 2 + 0.2 * 2 + 0.15 * 3 + 0.15 * 3 = 2.30 \tag{6}$$

Since 2.3 is less than 3, we have produced a shorter code, which saves space (and transmission time). How low can we go? Shannon proved that the expected number of bits needed to encode a message is lower bounded by the **entropy** of the probability distribution governing the data. (This is called the **source coding theorem**.) We discuss what we mean by entropy below.

2.2 Huffman coding*

There is a very simple and elegant algorithm for generating optimal symbol codes. The idea is to assign code words to symbols in the alphabet by building the binary tree up from the leaves. Start by simply taking the least two probable symbols in the alphabet and assinging them the longest codeword, which differ by 0 or 1; then merge these two symbols into a single symbol and repeat. Figure 2 gives an example.

In the example above, the Huffman code gives an expected number of bits of 2.30, whereas the entropy is $H = 2.2855$ bits (as we will see below). To achieve this lower bound requires that we give up the notion of using an integer number of bits per symbol; this results in what is called **arithmetic coding**. In practice this means we must encode group of symbols at a time (since we can't send fractional bits). See [Mac03] for details.

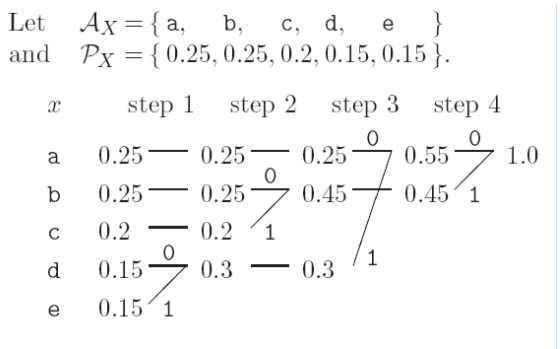
2.3 Entropy

Consider a discrete random variable $X \in \{1, \ldots, K\}$. Suppose we observe the event that $X = k$. We define the information content of this event as

$$h(k) = \log_2 1/p(X = k) = -\log_2 p(X = k) \tag{7}$$

The idea is that unlikely events (with low probability) convey more information. The **entropy** of a distribution p is defined as the average information content of a random variable X with distribution p:

$$H(X) = -\sum_{k=1}^{K} p(X = k) \log_2 p(X = k) \tag{8}$$



a_i	p_i	$h(p_i)$	l_i	$c(a_i)$
a	0.25	2.0	2	00
b	0.25	2.0	2	10
c	0.2	2.3	2	11
d	0.15	2.7	3	010
e	0.15	2.7	3	011

Figure 2: An example of Huffman coding. Source: [Mac03] p99.

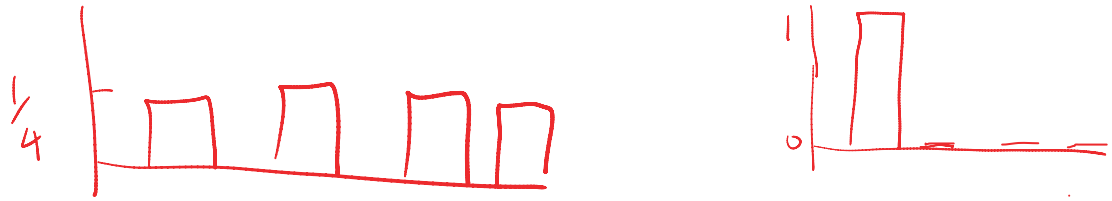


Figure 3: Some probability distributions on $X \in \{1, 2, 3, 4\}$. Left: a uniform distribution $p(x = k) = 1/4$. Right: a degenerate distribution $p(x) = 1$ if $x = 1$ and $p(x) = 0$ if $x \in \{2, 3, 4\}$.

For example, if $X \in \{1, \ldots, 5\}$ with distribution

$$p(a) = 0.25, p(b) = 0.25, p(c) = 0.2, p(d) = 0.15, p(e) = 0.15 \tag{9}$$

we find $H = 2.2855$. For a K-ary random variable, the entropy if maximized if $p(x = k) = 1/K$, i.e, the uniform distribution. In this case, $H(X) = \log_2 K$. The entropy is minimized ($H = 0$) if $p(x) = \delta(x - x^*)$ for some x^*, i.e., for a deterministic distribution. See Figure 3.

For the special case of binary random variables, $X \in \{0, 1\}$, we can write $p(X = 1) = \theta$ and $p(X = 0) = 1 - \theta$. Hence the entropy becomes

$$H(X) = -[p(X = 1) \log_2 p(X = 1) + p(X = 0) \log_2 p(X = 0)] \tag{10}$$

$$= -[\theta \log_2 \theta + (1 - \theta) \log_2 (1 - \theta)] \tag{11}$$

This is called the binary entropy function, and is also written $H(\theta)$, to emphasize that it is a function of the distribution (parameter) θ, rather than the random variable X. We plot this in Figure 4. We see that the maximum value is $H(X) = 1$ which occurs when the distribution is uniform $\theta = 0.5$:

$$-[\frac{1}{2} \log_2 \tfrac{1}{2} + (1 - \tfrac{1}{2}) \log_2 (1 - \tfrac{1}{2})] = -\log_2 \tfrac{1}{2} = 1 \tag{12}$$

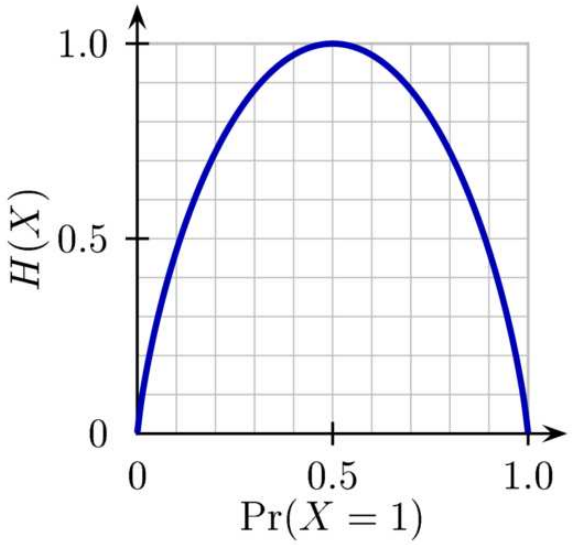


Figure 4: Entropy of a Bernoulli random variable as a function of $p(X = 1) = \theta$. The maximum entropy is $\log_2 2 = 1$. Source: `http://en.wikipedia.org/wiki/Information_entropy`.

2.4 Joint entropy

The joint entropy of two random variables X and Y is defined as

$$H(X, Y) = -\sum_{x,y} p(x, y) \log_2 p(x, y) \tag{13}$$

If X and Y are independent, then $H(X, Y) = H(X) + H(Y)$. In general, one can show (see Section 2.7.3) that

$$H(X, Y) \leq H(X) + H(Y) \tag{14}$$

For example, consider choosing an integer from 1 to 8, $n \in \{1, \ldots, 8\}$. Let $X(n)$ be the event that n is even, and $Y(n)$ be the event that n is prime.

n	1	2	3	4	5	6	7	8
X	0	1	0	1	0	1	0	1
Y	0	1	1	0	1	0	1	0

Clearly $p(X = 1) = p(X = 0) = 0.5$, so $H(X) = 1$; similarly $H(Y) = 1$. However, we will show that $H(X, Y) < H(X) + H(Y)$, since the events are not independent.

The joint probability distribution is

$p(X, Y)$	0	1
0	$\frac{1}{8}$	$\frac{3}{8}$
1	$\frac{3}{8}$	$\frac{1}{8}$

so the joint entropy is given by

$$H(X, Y) = -[\frac{1}{8} \log_2 \frac{1}{8} + \frac{3}{8} \log_2 \frac{3}{8} + \frac{3}{8} \log_2 \frac{3}{8} + \frac{1}{8} \log_2 \frac{1}{8}] = 1.8113 \tag{15}$$

So $H(X, Y) < H(X) + H(Y)$.

What is the lower bound on $H(X, Y)$? Clearly

$$H(X, Y) \geq H(X) \geq H(Y) \geq 0 \tag{16}$$

where $H(X, Y) = H(X)$ iff Y is a deterministic function of X. Intuitively this says that combining two systems can never reduce the overall uncertainty.

2.5 Conditional entropy

The **conditional entropy** of Y given X is the expected uncertainty we have in Y after seeing X:

$$\begin{aligned} H(Y|X) &\stackrel{\text{def}}{=} \sum_x p(x) H(Y|X=x) && (17)\\ &= -\sum_x p(x) \sum_y p(y|x) \log p(y|x) && (18)\\ &= -\sum_{x,y} p(x,y) \log p(y|x) && (19)\\ &= -\sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)} && (20)\\ &= -\sum_{x,y} p(x,y) \log p(x,y) - \sum_x p(x) \log \frac{1}{p(x)} && (21)\\ &= H(X,Y) - H(X) && (22) \end{aligned}$$

If X completely determines Y, then $H(Y|X) = 0$. If X and Y are independent, then $H(Y|X) = H(Y)$. Since $H(X,Y) \leq H(Y) + H(X)$, we have

$$H(Y|X) \leq H(Y) \quad (23)$$

with equality iff X and Y are independent. This shows that conditioning on data always decreases (or rather, never increases) ones uncertainty, *on average*.

2.6 Mutual information

The **mutual informatio**n between X and Y is how much our uncertainty about Y decreases when we observe X (or vice versa). It is defined as

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)\, p(y)} \quad (24)$$

It can be shown (exercise: just plug in the definitions) that this is equivalent to the following:

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) && (25)\\ &= H(Y) - H(Y|X) && (26) \end{aligned}$$

Subsituting $H(Y|X) = H(X,Y) = H(X)$ yields

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \quad (27)$$

Hence

$$\begin{aligned} H(X,Y) &= (H(X,Y) - H(Y)) + (H(X,Y) - H(X)) + (H(X) + H(Y) - H(X,Y)) && (28)\\ &= H(X|Y) + H(Y|X) + I(X,Y) && (29) \end{aligned}$$

See Figure 5.

Mutual information measures dependence between random variables in the following sense: $I(X,Y) \geq 0$ with equality iff $X \perp Y$. (The proof that $I(X,Y) = 0$ if X and Y are independent is easy; the proof that $I(X,Y) = 0 \implies X \perp Y$ is harder: see Section 2.7.3.) If $Y = X$, then the mutual information is maximal, and equal to $H(X) \leq \log_2 K$. Mutual information is similar in spirit to a **correlation coefficient**, but is much more general, because correlation only captures linear dependencies. Thus two variables may have a correlation coefficient of 0, even though they are (nonlinearly) related. However, their mutual information will never be zero in this case. (We will study correlation coefficients later in the context of linear regression.)

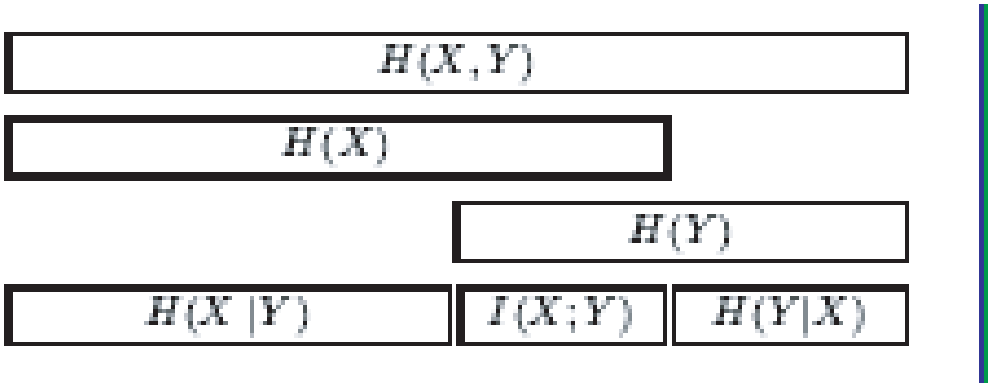


Figure 5: The relationship between joint entropy, marginal entropy, conditional entropy and mutual information.. Source: [Mac03].

Let us continue with the example above concerning prime and even numbers. Recall that $H(X) = H(Y) = 1$. The conditional distribution $p(Y|X)$ is given by normalizing each row:

$p(Y\|X)$	0	1
0	$\frac{1}{4}$	$\frac{3}{4}$
1	$\frac{3}{4}$	$\frac{1}{4}$

Hence the conditional entropy is

$$H(Y|X) = -[\frac{1}{8}\log_2\frac{1}{4} + \frac{3}{8}\log_2\frac{3}{4} + \frac{3}{8}\log_2\frac{3}{4} + \frac{1}{8}\log_2\frac{1}{4}] = 0.8113 \tag{30}$$

and the mutual information is

$$I(X,Y) = H(Y) - H(Y|X) = 1 - 0.8113 = 0.1887 \tag{31}$$

Note that it is sometimes useful to use the **conditional mutual information**, defined as

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z) \tag{32}$$

2.7 Relative entropy (KL divergence)

If X represents all possible (fixed-length) sentences in English, or all possible (fixed-sized) images, the true probability distribution $p(X)$ will be quite complicated. But the more accurate our model of this, call it $q(X)$, the fewer the number of bits we will need to encode data from this source. (Hence learning better models of data results in better codes.) The quality of our approximation is captured in the notion of **Kullback-Leibler (KL) divergence**, also called **relative entropy**.

The KL divergence is very important "distance" measure between two distributions, p and q. It is defined as follows

$$\mathcal{D}(p||q) \stackrel{\text{def}}{=} \sum_k p_k \log \frac{p_k}{q_k} \tag{33}$$

It is not strictly a distance, since it is asymmetric. The KL can be rewritten as

$$\mathcal{D}(p||q) = \sum_k p_k \log p_k - \sum_k p_k \log q_k = -\sum_k p_k \log q_k - H(p) \tag{34}$$

where $\sum_k p_k \log q_k$ is called the **cross entropy**. This makes it clear that the KL measures the extra number of bits we would need to use to encode X if we thought the distribution was q but it was actually p.

2.7.1 Minimizing KL divergence to the empirical distribution is maximizing log likelihood

Since $D(p||q)$ measures the distance between the true distribution p and our approximation q, we would like to minimize this. Let p be the empirical distribution of the data D

$$p(x) = \frac{1}{N} I(x \in D) = \frac{1}{N} \sum_{n=1}^{N} \delta(x - x_n) \tag{35}$$

which assigns mass $1/N$ if x equals one of the training points x_i, and 0 mass otherwise. Then the KL becomes

$$KL(p||q) = \sum_{x \in D} \frac{1}{N} \log \frac{1/N}{q(x)} = -\frac{1}{N} \sum_{x \in D} \log q(x) + const \tag{36}$$

Hence

$$\hat{q} = \arg\min_q KL(p||q) = \arg\max_q \frac{1}{N} \sum_n \log q(x_n) \tag{37}$$

In otherwords, the distribution that minimizes the KL to the empirical distribution is the maximum likelihood distribution. In practice, instead of optimizing the function q, we optimize its parameters θ:

$$\hat{\theta} = \arg\max_\theta \frac{1}{N} \sum_n \log q(x_n|\theta) \tag{38}$$

2.7.2 Mutual information as KL divergence

The KL can be used to compare a joint distribution with a factored distribution. This quantity is called the **mutual information** between X and Y, and is defined as

$$\mathcal{I}(X, Y) \stackrel{\text{def}}{=} \mathcal{D}(P(X, Y)||P(X)P(Y)) \tag{39}$$

It is easy to show (exercise) that this gives the same results as before, namely

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X, Y) \tag{40}$$

2.7.3 KL is always non-negative *

KL satisfies $D(p||q) \geq 0$ with equality iff $p = q$. The fact that $D(p||q) = 0$ if $p = q$ is easy to see, since we have terms of the form $\log p(x)/q(x) = \log 1 = 0$. We will now show KL is always positive.

Recall that a **concave** function f is one which lies above any chord

$$f(\lambda x_1 + (1 - \lambda)x_2) \geq \lambda f(x_1) + (1 - \lambda) f(x_2) \tag{41}$$

where $0 \leq \lambda \leq 1$. Intuitively this an inverted bowl: see Figure 6. A function f is **convex** if $-f$ is concave (a U bowl).

Jensen's inequality states that, for any **concave function** f,

$$E[f(X)] \leq f(E[X]) \tag{42}$$

i.e.,

$$\sum_x p(x) f(x) \leq f(\sum_x p(x)) \tag{43}$$

This can be proved by induction by setting $\lambda = p(x = 1)$ and $1 - \lambda = \sum_{x=2}^{K} p(x)$; the base case uses the definition of concavity.

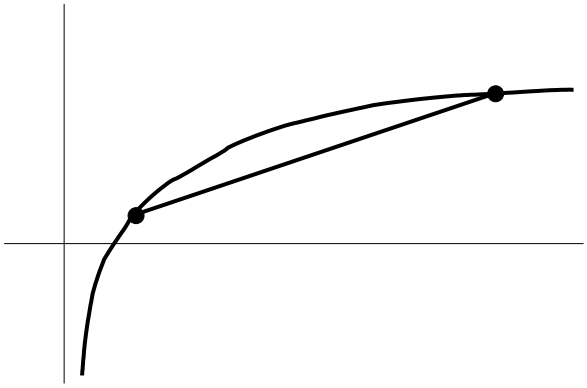


Figure 6: An illustration of a concave function ($\log x$).

To prove that $D(p||q) \geq 0$, let $u(x) = p(x)/q(x)$ and $f(u) = \log 1/u$ be a convex function. Then

$$\begin{aligned} D(p||q) &= E[f(q(x)/p(x))] && (44) \\ &\geq f\left(\sum_x p(x)\frac{q(x)}{p(x)}\right) && (45) \\ &= \log(\frac{1}{\sum_x q(x)}) = 0 && (46) \end{aligned}$$

From this it follows that

$$I(X,Y) = D(p(x,y)||p(x)p(y)) \geq 0 \quad (47)$$

and hence

$$H(X) + H(Y) \geq H(X,Y) \quad (48)$$

as claimed above.

2.7.4 Forward and reverse KL divergences *

If $q_k = 0$ then $D(p||q) = \infty$ unless $p_k = 0$ also. Hence to minimize $D(p||q)$, q should "cover" p. For example, if p is the empirical distribution of the training set, then q should not assign zero probability to anything in the training set. When studying variational inference, it is more common to optimize $D(q||p)$ with respect to q, since this is computationally cheaper (by assumption we can take expectations wrt q but not wrt p). But to minimize $D(q||p)$, we need $q_k = 0$ whenever $p_k = 0$; hence q should be "under" one of p's modes. See Figures 7 and 8 for an illustration of these differences, which will become important later in the book.

Note that finding a distribution q to minimize $D(p||q)$ is hard, since it requires computing expectations wrt p, which by assumption is a complex distribution (otherwise we wouldn't need to approximate it). Finding q to minimize the "reverse" KL $D(q||p)$ is relatively straightforward, however.

2.8 Information theoretic quantities for continuous data *

If X is a continuous random variable with pdf $p(x)$, we define the **differential entropy** as

$$h(X) = -\int_S p(x)\log p(x)dx \quad (49)$$

where S is the support of the random variable. (We assume this integral exists.)

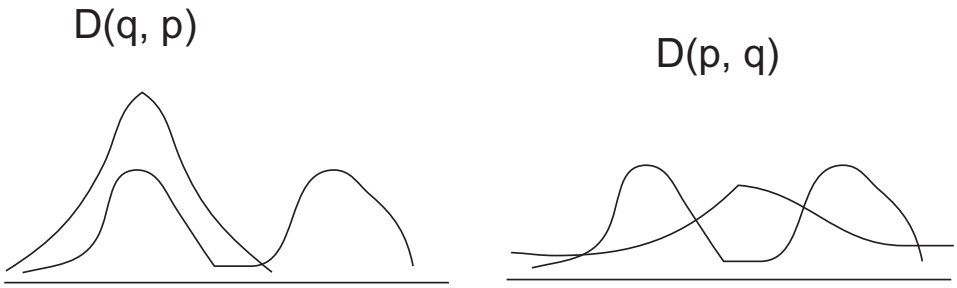


Figure 7: Left: Minimizing $D(q||p)$ picks one of the modes of p. Right: minimizing $D(p||q)$ tries to globally "cover" p.

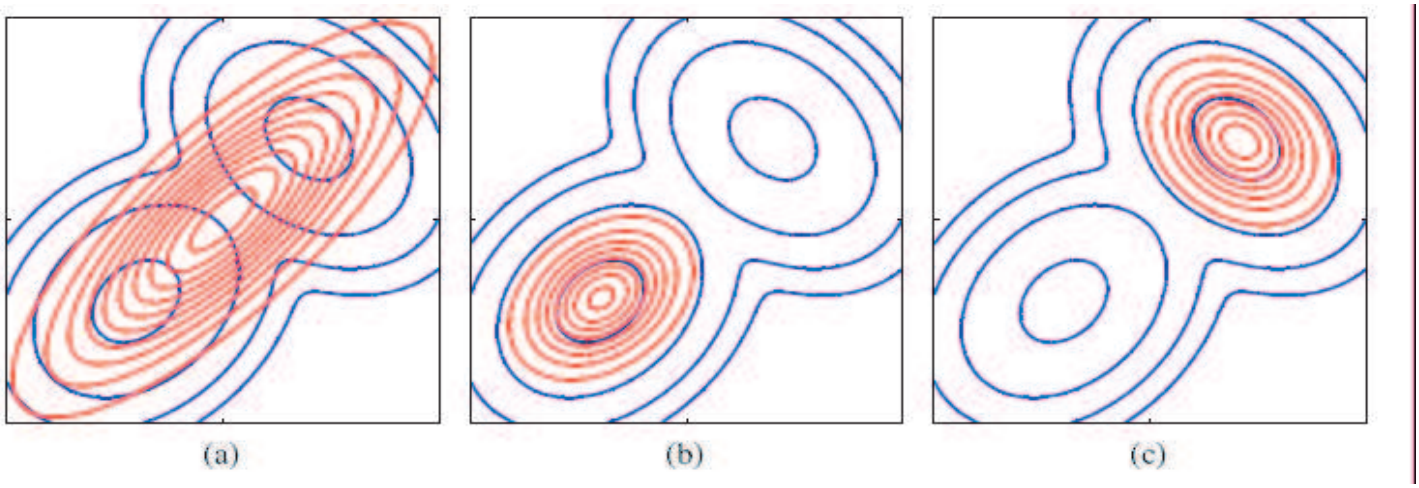


Figure 8: Blue: the true distribution p is a mixture of 2 Gaussians. Red: the approximating distribution q. Left: minimizing $D(p||q)$ leads to a broad distribution. Middle: minimizing $D(q||p)$ picks the bottom mode; Right: minimizing $D(q||p)$ can also pick the top mode. Source: [Bis06] Fig 5.4.

For example, suppose $X \sim U(0, a)$. Then

$$h(X) = -\int_0^a \frac{1}{a} \log \frac{1}{a} dx = \log a \tag{50}$$

Unlike the discrete case, differential entropy can be negative: if $a < 1$, we have $h(X) < 0$.

As another example, suppose $X \sim \mathcal{N}(0, \sigma^2)$, so

$$p(x) = (1/\sqrt{2\pi\sigma^2}) \exp(-x^2/2\sigma^2) \tag{51}$$

Hence the differential entropy is

$$\begin{aligned} h(X) &= -\int_{-\infty}^{\infty} p(x) && (52)\\ &= -\int p(x)[-\frac{x^2}{2\sigma^2} - \log\sqrt{2\pi\sigma^2}] && (53)\\ &= \frac{EX^2}{2\sigma^2} + \frac{1}{2}\log 2\pi\sigma^2 && (54)\\ &= \frac{1}{2} + \frac{1}{2}\log 2\pi\sigma^2 && (55)\\ &= \frac{1}{2}\log e + \frac{1}{2}\log 2\pi\sigma^2 && (56)\\ &= \frac{1}{2}\log 2\pi e\sigma^2 && (57)\\ &= \log(\sigma\sqrt{2\pi e}) && (58) \end{aligned}$$

If we use log base e, the units are called "nats". If we use log base 2, the units are called bits. A list of differential entropies for various univariate distributions can be found at `http://en.wikipedia.org/wiki/Differential_entropy`. However, in general, for non-standard distributions, especially multivariate ones, the most common approach is to **discretize (quantize)** the data (we discuss techniques to do this later), and then use discrete entropy. (See [LM04] for an interesting alternative that bypasses the density estimation step.)

2.9 Entropy rates of a stochastic process *

A **stochastic process** is an indexed sequence of random variables $p(X_1, \ldots, X_D)$. A Markov chain is a simple example. In this case,

$$p(X_1, \ldots, X_D) = p(X_1) \prod_{i=2}^{D} p(X_i|X_{i-1}) \tag{59}$$

The **entropy rate** of a stochastic process $\{X_i\}$ is a measure of its predictability, and is defined as

$$H(\mathcal{X}) = \lim_{D\to\infty} \frac{1}{D} H(X_1, \ldots, X_D) \tag{60}$$

when the limit exists. For example, suppose $X_1, \ldots, X_D$ are iid. Then

$$H(\mathcal{X}) = \lim_{D\to\infty} \frac{1}{D} H(X_1, \ldots, X_D) = \frac{DH(X_1)}{D} = H(X_1) \tag{61}$$

which is just the entropy rate per symbol.

An alternative definition of the entropy rate is

$$H'(\mathcal{X}) = \lim_{D\to\infty} H(X_D|X_1, \ldots, X_{D-1}) \tag{62}$$

One can show that for a stationary stochastic process, the limits in both definitions exist and are equal, i.e., $H(\mathcal{X}) = H'(\mathcal{X})$ (see [CT91]). For example, suppose $\{X_i\}$ is a stationary Markov chain with stationary distribution π and transition matrix T. Then the entropy rate is

$$\begin{aligned} H(\mathcal{X}) &= \lim H(X_D|X_{D-1}, \ldots, X_D) && (63)\\ &= \lim H(X_D|X_{D-1}) && (64)\\ &= H(X_2|X_1) && (65)\\ &= \sum_j p(X_1 = j) H(X_2|X_1 = j) && (66)\\ &= \sum_j \pi(j)[-\sum_k p(X_2 = k|X_1 = j) \log p(X_2 = k|X_1 = j)] && (67)\\ &= -\sum_j \pi(j) \sum_k T(j,k) \log T(j,k) && (68) \end{aligned}$$

For example, for a 2 state chain with transition matrix

$$T = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix} \tag{69}$$

and stationary distribution

$$\pi_1 = \frac{\beta}{\alpha+\beta}, \;\; \pi_2 = \frac{\alpha}{\alpha+\beta}, \tag{70}$$

we have that the entropy rate is

$$H(\mathcal{X}) = \frac{\beta}{\alpha+\beta} H(\alpha) + \frac{\alpha}{\alpha+\beta} H(\beta) \tag{71}$$

where $H(\alpha)$ is the binary entropy function.

3 Applications of information theory in machine learning

3.1 Model selection using minimum description length (MDL)

To losslessly send a message about an event x with probability $p(x)$ takes $L(x) = -\log_2 p(x)$ bits. Suppose instead of sending the raw data, you send a model H and then send the residual errors (the parts of the data not predicted by the model). This takes $L(D, H)$ bits:

$$L(D, H) = -\log p(H) - \log p(D|H) \tag{72}$$

The best model is the one with the overall shortest message. This is called the **minimum description length (MDL)** principle, and is essentially the same as **Occam's razor**. See Figure 9 for an illustration, and [RY00] for more details.

Note that MDL is essentially equivalent to MAP estimation (which is an approximation to full Bayesian inference), since the MAP model is given by the one with maximum (log) posterior

$$\log p(h|D) = \log p(h) + \log p(D|h) + \text{const} \tag{73}$$

Since there is a 1:1 mapping between coding length and probabilities, choosing between the MDL approach and the Bayesian approach it is mostly a question of convenience. For example, sometimes it is easier to think of the cost of encoding a model than to define a prior on models. However, in the Bayesian approach one can perform Bayesian model averaging, which, in terms of predictive accuracy, is always better than (or at least as good as) picking the single best model.

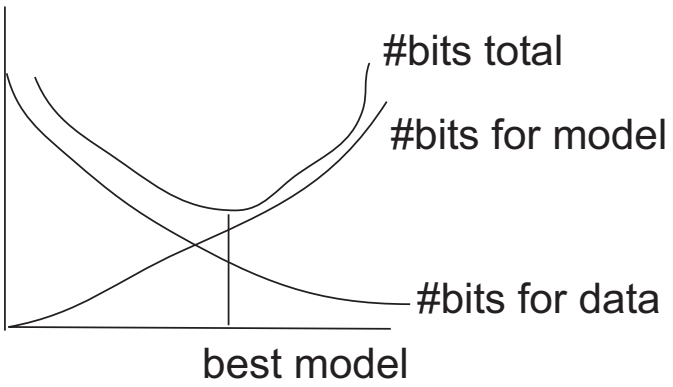


Figure 9: An illustration of the MDL principle. We plot number of bits versus model complexity. The model for which the total number of bits, $L(D, H)$ is minimal, is assumed to have lowest generalization error.

Figure 10: An illustration of active learning. Left: current information state, with A, B, C and D unlabeled. Right: possible "true" state, where the positives are "sandwiched" between negative examples.

3.2 Active learning

Consider the problem of learning a classifier. Suppose some of the data points are labeled, but others are unlabeled. (This is an example of **semi-supervised learning**.) Now suppose we are allowed to ask for the labels of one or more points. (This is called **active learning**.) Which points should we query?

Consider the example in Figure 10(left). Suppose we can choose to get the label for points A,B,C or D. Intuitively, it seems likely that A is positive (since its **nearest neighbors** are positive) and that C is negative. We might imagine there is a "dividing line" (separating hyper-plane) between the positive and negative points, in which case B would be positive and D negative. In this case, we don't need to ask for more labels, we have correctly learned the "concept".

But what if the true situation is more like Figure 10(right), where the positives are "sandwiched" between the negatives? If this kind of hypothesis is also in our hypothesis class (i.e., we entertain the possibility that the data could be separated into +-, or could be sandwiched -+-), then we will be very uncertain about the labels of B, since our prediction will be a mixture of these two hypotheses. Hence the entropy of our prediction at B will be higher than at any other point:

$$H(p(y|x = B, \mathcal{D})) \geq H(p(y|x', \mathcal{D})), x' \in \{A, C, D\} \tag{74}$$

where $p(y|x = B, \mathcal{D})$ is the probability distribution over labels at location B given the training data $\mathcal{D}$. (Arguably we are equally uncertain about D's label.) Hence a reasonable heuristic is to query the points whose entropy is highest.

3.3 Feature selection

In many classification and regression problems, where the goal is to compute $p(y|x)$, the input $x_{1:p}$ may be a high dimensional vector. Not all of the components (dimensions) of x may be relevant for predicting y. Feature selection is the task of finding the relevant components. This results in a simpler model that may be easier to understand and may even perform better than using all p features.

There are two main approaches to feature selection (see [GE03] for a good review): **filter** methods preprocess the features x_i and then build a classifier using the k relevant features, $p(y|x_{1:k})$; **wrapper** methods try all subsets $p(y|x_s)$ and pick the subset that performs the best. It is clear that filter methods are much more comptuationally efficient, since they only have to train the classifier once. A common measure of relevance is mutual information: we compute $I(X_i, Y)$ for each feature X_i and keep the k features with highest mutual information.

For example, referring to Figure 10, each input can be described by its horizontal and vertical coordinate; call these components 1 and 2. It seems clear that the vertical coordinate is irrelevant for predicting the class label Y. Hence we would expect to find $I(X_1, Y) \gg I(X_2, Y)$.

As another example, consider a naive Bayes classifier with C classes and binary features, $x_i \in \{0, 1\}$. Let $\pi_c = p(y = c)$, $\theta_{ic} = p(x_i = 1|y = c)$ and

$$\theta_i = p(x_i = 1) = \sum_c p(x_i = 1|y = c)p(y = c) = \sum_c \theta_{ic}\pi_c \tag{75}$$

The mutual information between feature i and the class label is given by

$$\begin{aligned}
I_i &= \sum_{x=0}^{1}\sum_{c=1}^{C} p(X_i = x, y = c) \log \frac{p(X_i = x|y = c)p(y = c)}{p(X_i = x)p(y = c)} && (76)\\
&= \sum_{x=0}^{1}\sum_{c} p(X_i = x|y = c)p(y = c) \log \frac{p(X_i = x|y = c)}{p(X_i = x)} && (77)\\
&= \sum_{c} p(X_i = 1|y = c)p(y = c) \log \frac{p(X_i = 1|y = c)}{p(X_i = 1)} + && (78)\\
&\quad \sum_{c} p(X_i = 0|y = c)p(y = c) \log \frac{p(X_i = 0|y = c)}{p(X_i = 0)} && (79)\\
&= \sum_{c} \left[\theta_{ic}\pi_c \log \frac{\theta_{ic}}{\theta_i} + (1 - \theta_{ic})\pi_c \log \frac{1 - \theta_{ic}}{1 - \theta_i}\right] && (80)
\end{aligned}$$

This can be used to select relevant features before fitting the naive Bayes model.

References

[Bis06] C. Bishop. *Pattern recognition and machine learning*. Springer, 2006. Draft version 1.21.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. of Machine Learning Research*, 3:1157–1182, 2003.

[LM04] Erik Learned-Miller. Hyperspacings and the estimation of information theoretic quantities. Technical Report 04-104, U. Mass. Amherst Comp. Sci. Dept, 2004.

[Mac03] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[RY00] Jorma Rissanen and Bin Yu. Coding and compression: a happy union of theory and practice. *J. of the Am. Stat. Assoc.*, 95:986–988, 2000.